

plurality of users. In a case where a large number of users accesses the resource program, i.e. in a case where $N > M$, this alternative efficiently uses available storage space.

Thus, the process depicted in Fig. 4 provides internationalization support to applications which are written to be independent of a language or format desired by a particular user. Functions are provided that support provision of localized text resources, that convert locally different user input like date, time, and floating point representations into resource identifiers that are intended to store such data, and that convert resource identifiers, such as a current date, using resource functions into a localized format.

Turning to Fig. 5a, that figure illustrates a flow chart of a process performed by the resource program in accordance with another embodiment consistent with methods, systems, and articles of manufacture consistent with the present invention. The illustrated process outlines the steps for performing the lookup operation for retrieving resource data based on a resource identifier for one or a plurality of users.

The process illustrated in Fig. 5a may be performed using the data processing system 100 depicted in Fig. 1, however, the process of Fig. 3a is not limited thereto.

Since the steps for setting a user parameter (step 21), executing an application (step 22), reading a resource identifier (step 23), and loading a lookup object (step 24) are similar to the steps described with reference to Fig. 4, description starts at an entry point S24, that is, after step 24 described with reference to Fig. 4.

Referring to Fig. 5a, in step S3a1, the lookup component 183 generates a string identifier. The string identifier consists of the read resource identifier and the user parameter. The lookup component 183 may generate the string identifier in a calculation operation that combines the resource identifier and the user parameter.

In step S3a2, the lookup component 183 then performs a lookup operation using the user parameter and the loaded lookup object. In this case, the lookup object includes an assignment between the string identifier and the resource data

depending on the resource identifier. Accordingly, the lookup component 183 first converts the resource identifier into the string identifier using the user parameter, and then retrieves the corresponding resource data, for example, from the lookup table 188.

The resource program can then provide the read resource data that depends on the user parameter to the user.

As an example, it is assumed that a resource identifier `resID_IP`, which is associated with the expression "Intellectual Property", is read by the application. Further, it is assumed that there is a user parameter "49", which identifies the user's environment as "Germany", and a user parameter "01", which identifies the user's environment as "United States".

Accordingly, when the user parameter component 181 receives an input from a user to set the user parameter "49", the lookup component 183 generates a string identifier that consists of the resource identifier `resID_IP` and the user parameter "49". The generated string identifier may read as `resID_IP_49`. This string identifier is preferably stored by the lookup component 181 in the lookup object in association with the German expression "Gewerblicher Rechtsschutz". When the lookup function for obtaining from this lookup object is called, based on the user parameter "49" and the resource identifier `resID_IP`, the lookup component 183 retrieves the resource data "Gewerblicher Rechtsschutz" for presentation to the user.

When the user parameter component 181 receives an input from a user to set the user parameter "01", the lookup component 183 correspondingly generates a string identifier consisting of the resource identifier `resID_IP` and the user parameter "01". The generated string identifier may read as `resID_IP_01`. Similar to the German version, this string identifier is preferably stored by the lookup component 181 in the lookup object in association with the English expression "Intellectual Property". When the lookup function for obtaining from this lookup object is called, based on the user parameter "01" and the resource identifier `resID_IP`, the lookup component 183 retrieves the resource data "Intellectual Property" for presentation to the user.

Accordingly, in the above-described illustrative example, the lookup object may contain the information shown in Table 1.

resID_IP_49	"Gewerblicher Rechtsschutz"
resID_IP_01	"Intellectual Property"

Table 1

In another example, the resource program presents a floating point number to the user, based on a resource function.

In this illustrative example, it is assumed that a floating point number 1fpt7 is to be presented to a user, wherein fpt stands for floating point. The number to be presented to the user is "1.7". Further, it is again assumed that the user parameter "49" stands for the user environment "Germany" and the user parameter "01" stands for the user environment "United States".

When the user parameter component 181 receives an input from a user to set the user parameter "49", the lookup component 183 generates a string identifier that consists of the resource identifier fpt and the user parameter "49". The generated string identifier may read as fpt_49. The lookup object thus preferably maintains a rule for placing a "," in a floating point number in association with the string identifier fpt_49. Accordingly, after the lookup component 183 retrieves the rule stored in association with the string identifier fpt_49, the application component 182 presents the number 1fpt7 the user. Based on the selected user parameter "49", the presented number will be "1,7".

Similarly, when the user parameter component 181 receives an input from a user to set the user parameter "01", the lookup component 183 generates a string identifier that consists of the resource identifier fpt and the user parameter "01". The generated string identifier may read as fpt_01. The lookup object thus preferably maintains a rule for placing a "." in a floating point number in association with the string identifier fpt_01. Accordingly, after the lookup component 183 retrieves the rule stored in association with the string identifier